

Apuntes de Algoritmos Genéticos

Parte V: Un ejemplo

M. Angélica Pinninghoff J. - Ricardo Contreras A.
Magíster en Inteligencia Artificial
Universidad Adolfo Ibáñez

2022

1. Introducción

Haremos la suposición de que los conceptos previos han sido ya estudiados y comprendidos. Eso significa que estamos preparados para enfrentar la resolución de un problema concreto. Lo importante de este punto es darse cuenta que, una vez comprendida la estructura y operación de los algoritmos genéticos, es posible generalizar a una gama casi infinita de problemas, con la indicación que las funciones de evaluación de cada problema, dependerán de la naturaleza del mismo.

2. Un Ejemplo de Uso.

Con la intención de aclarar un poco más los conceptos cubiertos hasta aquí, se presenta una aplicación de algoritmos genéticos a un problema de optimización, tomado (y adaptado) de Kaufmann [7].

Un grupo de financieras ha resuelto invertir 10 millones de pesos en la promoción de una nueva marca de vino: *Vinacho*. Así pues, en cuatro ciudades se decide iniciar una campaña comercial: Antofagasta en el norte, Valparaíso en el centro, Rancagua en el centro y Concepción en el sur. A esas cuatro ciudades van a corresponder las zonas comerciales I, II, III y IV. Se ha realizado un estudio de mercado en cada una de las zonas citadas y se han establecido curvas de ganancia medias, en millones de pesos, en función de las inversiones totales (almacenes, tiendas de venta, representantes, publicidad, etc.). Estos datos se ilustran en la tabla a continuación. Para



Figura 1: El origen del producto.

simplificar los cálculos supondremos que las asignaciones de créditos o de inversiones deben hacerse por unidades de 1 millón de pesos. La pregunta es ¿en dónde se deben asignar los 10 millones de pesos de los que se dispone para que la ganancia total sea máxima?

<i>Inversión</i>	<i>Beneficio I</i>	<i>Beneficio II</i>	<i>Beneficio III</i>	<i>Beneficio IV</i>
0	0	0	0	0
1	0.28	0.25	0.15	0.20
2	0.45	0.41	0.25	0.33
3	0.65	0.55	0.40	0.42
4	0.78	0.65	0.50	0.48
5	0.90	0.75	0.62	0.53
6	1.02	0.80	0.73	0.56
7	1.13	0.85	0.82	0.58
8	1.23	0.88	0.90	0.60
9	1.32	0.90	0.96	0.60
10	1.38	0.90	1.00	0.60



Figura 2: El producto en cuestión.

2.1. Representación.

Para poder aplicar el algoritmo genético, lo primero que necesitamos determinar es cuál será el esquema a utilizar para representar las posibles soluciones al problema. En este caso, si optamos por la representación binaria necesitamos 4 bits para representar cada solución, porque cada una admite 11 valores (de 0 a 10). Como existen 4 valores independientes (uno por cada zona de estudio), se requieren entonces 16 bits ($4 * 4$) por cada cromosoma. Es importante hacer notar que se requiere una función de codificación (i.e., que transforme el valor de la inversión a código binario) y una de decodificación (que realice el proceso inverso).

Debido a que en este caso los 4 bits utilizados para representar una solución pueden producir más valores de los que se necesitan, se usará una función de ajuste que haga que los resultados producidos siempre se encuentren en el rango válido.

2.2. Función de Aptitud.

Dado que el objetivo es obtener inversiones que totalicen 10 millones, y que tengan un beneficio máximo, podemos usar la siguiente función de aptitud penalizada:

$$F(x) = \frac{C_1 + C_2 + C_3 + C_4}{500*V + 1}$$

Donde C_1, C_2, C_3, C_4 son las ganancias por zona, que se calculan de acuerdo a los valores entregados por la tabla anterior, y V es el valor absoluto de la diferencia entre la suma obtenida de las inversiones y 10. Nótese que cuando no se viole ninguna restricción, es decir cuando la suma de inversiones sea exactamente 10, la función de aptitud no será penalizada.

2.3. Operadores.

Se usará un cruzamiento de dos puntos. La probabilidad que se dará a la misma será del 80 %. En cuanto a la mutación, se le asignará una probabilidad del 1 %. El tamaño de población para este ejemplo será de 50 cromosomas, y se dejará al sistema ejecutarse hasta completar 20 generaciones.

2.4. Resultados.

El resultado obtenido en una ejecución típica es un beneficio de 1.81 millones, correspondientes a invertir 4 millones en la zona comercial I, 3 millones en la zona comercial II, 1 millón en la zona III y 2 millones en la zona IV. Esta es la solución óptima, la cual se obtuvo originalmente mediante programación dinámica [7]. Para el algoritmo genético, ejecutando en un PC convencional, el tiempo para encontrar esta solución llegó a los 13 segundos. Lo importante de destacar sin embargo es que en este caso, si deseáramos analizar inversiones que sumen otra cantidad, y en unidades menores al millón, el algoritmo genético debería modificarse de manera mínima, mientras que la programación dinámica requeriría una cantidad tal de trabajo que prácticamente se volvería inoperante.

Aunque éste es sólo un ejemplo sencillo del uso de algoritmos genéticos para resolver problemas de optimización con restricciones, puede alcanzar a percibirse el poder de la técnica en comparación con métodos tradicionales de búsqueda. Para interiorizarse más sobre las múltiples aplicaciones de algoritmos genéticos, se recomienda consultar [5, 1, 3]. Para una mayor explicación sobre los detalles de implementación elementales, resulta de utilidad analizar el SGA (Simple Genetic Algorithm) incluido en [5]. Este programa está escrito, en una versión de dominio público en C.



Figura 3: El resultado...

Referencias

- [1] Booker, L. B., Goldberg, D. E., Holland, J. H. *Classifier Systems and Genetic Algorithms*. Artificial Intelligence, 40, 1989, pp. 235-282.
- [2] Buckless, H. P., Petri, F. E. *Genetic Algorithms* . IEEE Computer Society Press, 1992.
- [3] Davis, L. (Ed.) *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, 1991.
- [4] Forrest, S. *Genetic Algorithms: Principles of Natural Selection Applied to Computation*. Science, Vol. 261, N. 5123, August, 1993.
- [5] Goldberg, E. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, 1989.
- [6] Holland, J. H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [7] Kaufmann, A., Faure, R. *Invitación a la Investigación de Operaciones*. Cecsa, Mexico, 1977.
- [8] Koza, J. R. *Genetic Programming. On the Programming of Computers by Means of Natural Selection*. The MIT Press, 1992.
- [9] Koza, J. R. *Genetic Programming II: Automatic Discovery of Reusable Programs*. The MIT Press, 1992.

- [10] Porter, K. *Handling Huge Arrays*. Dr. Dobb's Journal of Software Tools for the Professional Programmer. Vol. 13, N. 3, 1988.
- [11] Rawlins, G. J. E. *Foundations of Genetic Algorithms*. Morgan Kaufmann Publishers, 1993.
- [12] Ribeiro Filho, J. L., Treleaven, Ph. C., Alippi, C. *Genetic Algorithm Programming Environments*. IEEE Computer, June, 1994.
- [13] Smith, R. E., Goldberg, D. E., Earickson, J. A. *SGA-C: A C-language Implementation of a Simple Genetic Algorithm*. TCGA Report N. 91002, The Clearinghouse for Genetic Algorithms, The University of Alabama, May, 1991.
- [14] Srinivas, M., Patnaik, L. M. *Genetic Algorithms: A Survey*. IEEE Computer, June, 1994.
- [15] Whitley, L. D. *Foundations of Genetic Algorithms*. Morgan Kaufmann Publishers, 1993.